情報工学実験１
ーディジタル回路ー
基本ゲート回路

075730G:澤嶋千明

提出締切日:2008/06/27

共同研究者
075711A:山口藍
075726J:洌鎌温子
075759E:真地泰子

実験目的

現代社会に欠かすことのできないコンピュータは、大規模なディジタル回路として構成されている。本実験では、ディジタル回路の構成要素である基本ゲート回路と論理演算の基礎を習得することを目的とする。また本実験ではNANDゲートを用いて他のゲート回路（NOT、AND、OR、NOR、XOR）を構成することによって、汎用ロジックICおよびブレッドボード、直流電源などの基本的な使用法についても学ぶ。

実験概要

この実験はこれまでに習ってきた複数の回路を実際に組み、その動作を確認する旨のものである。
まずは実際に回路を組む前に、NANDゲートを用いたNOT、AND、OR、XOR、XORそれぞれの回路図を作成する。この回路図からゲート回路を組むので解り易く作成しておく。
次にブレッドボードとをNANDゲートICを用いてゲート回路を作成する。これはさきほど作成しておいたそれぞれの回路図から組み立てる。その回路図が正しいものか確認する為である。出力はLEDに繋げ、結果が解る様にしておく。
作成したら直流電源を使い電流を流す。このときのLEDの発光より、それぞれの回路が当たっているのか確認する。

実験

(1) NANDゲートのみを用いて、NOT、AND、OR、NOR、XORゲートを設計せよ(NANDゲートのみで構成された回路図を描け)。

(2) 実験(1)で設計した各ゲート回路を実際にNANDゲートICを用いてブレッドボード上に実現し、それらの動作を確認せよ。

実験結果

実験(1)の結果について

- NANDゲートのみを用いて実現された、NOT、AND、OR、NOR、XORゲートのそれぞれの回路を示せ。
  NANDゲートのみを用いて実現されたそれぞれの回路を、NOT回路(回路図:図1、真理値表:表1)、AND回路(回路図:図2、真理値表:表2)、OR回路(回路図:図3、真理値表:表3)、NOR回路(回路図;図4、真理値表:表4)、XOR回路(回路図:図5、真理値表:表5)として示しておく。

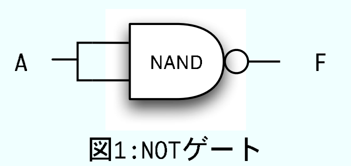

図1:NOTゲート

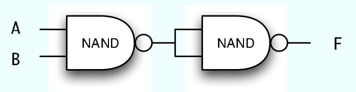

図2:ANDゲート

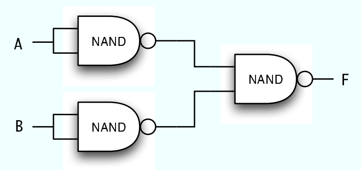

図3:ORゲート

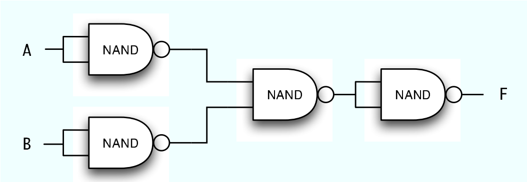

図4:NORゲート

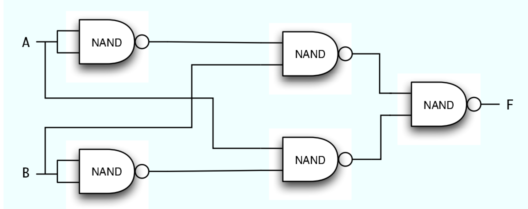

図5:XORゲート

A	f
0	1
1	0

表 1: NOT の真理値表

A	B	f
0	0	0
0	1	0
1	0	0
1	1	1

表 2: AND の真理値表

A	B	f
0	0	0
0	1	1
1	0	1
1	1	1

表 3: OR の真理値表

A	B	f
0	0	1
0	1	0
1	0	0
1	1	0

表 4: NOR の真理値表

A	B	f
0	0	0
0	1	1
1	0	1
1	1	0

表 5: XOR の真理値表

- 上記の各回路図が、それぞれ NOT、AND、OR、NOR、XOR ゲートの機能を実現していることを説明せよ。

  上記の各回路図と並記したそれぞれの真理値表より、それぞれ NOT、AND、OR、NOR、XOR ゲートと同じであるので同等の機能を実現していると言える。

- 上記の各回路図が導かれた経路を説明せよ。

  NAND の論理式は $f = \overline{A \cdot B}$ である。

  - **NOT**

    否定の論理式は $f = \overline{A}$ であるので、$f = \overline{A \cdot A}$ で表す事ができる。
    つまり、NAND ゲートの 2 入力を一つの入力として捉えればよい。

  - **AND**

    論理積の論理式は $f = A \cdot B = AB$ であるので、
    ド・モルガンの定理より $f = \overline{\overline{A \cdot B}}$ と表すことができる。二重否定すればよいので、A と B の入力の後、NOT 回路をつなげればよい。

  - **OR**

    論理和の論理式は $f = A + B$ であるので、
    ド・モルガンの定理より $f = \overline{\overline{A} \cdot \overline{B}}$ と表すことができる。入力された A と B の否定をさらに否定すればよい。

  - **NOR**

    論理和否定の論理式は $f = \overline{A + B}$ であるので、
    ド・モルガンの定理より $f = \overline{\overline{\overline{A} \cdot \overline{B}}}$ と表すことができる。論理和、OR 回路を否定すればよい。

  - **XOR**

    排他的論理和の論理式は $f = A \cdot \overline{B} + \overline{A} \cdot B = A \oplus B$ であるので

ド・モルガンの定理より $f = \overline{\overline{A \cdot \overline{B}} \cdot \overline{\overline{A} \cdot B}}$ と表せられる。
つまり、A、B それぞれの入力から否定したものとそうでないものを取り、$A \cdot \overline{B}$、$\overline{A} \cdot B$ を取ってからこれらを NOT 回路に繋げばよい。

実験 (2) の結果について

- NAND ゲート IC を用いてブレッドボード上に実現した各回路が、それぞれ、NOT、AND、OR、NOR、XOR ゲートと同様な動作をしたか否かを報告せよ
  各回路ともそれぞれ、NOT、AND、OR、NOR、XOR ゲートと同様な動作をすることを実験で確認した。

考察

実験 (1) の考察について

- 上記の実験結果で報告した回路よりも、使用する NAND 回路ゲートの数を減らすことが可能かどうかを、NOT、AND、OR、NOR、XOR ゲートのそれぞれについて考察せよ。
  それぞれの回路、NOT、AND、OR、NOR、XOR ゲートはどれも最小の数の NAND 回路ゲートで作られている。

実験 (2) の考察について

- ブレッドボード上で回路を構成する際に、配線ミスを減らすためには、配線数そのものを減らすことが望ましい。配線数を減らす為の工夫について考察せよ。
  配線数を減らす為には、それに使われるゲート数を減らさなければならない。これを減らす為、しっかり計算し設計する必要がある。
- 配線数を減らす以外の配線ミスを減らす為の工夫について考察せよ。
  上記に書いた以外の方法としては、隣合った配線には短めのものを使うことで空間的に余裕を持たせる。また、それぞれの入出力で色を統一し、色でその配線の意味が一目でわかるような工夫がよいと考えられる。

その他の考察について

- 本実験を通して得られた新たな知見について詳しく説明せよ。

  初めてブレッドボードに触れましたが、使い方が理解することができた。電線配線部は横に繋がっており、部品配線部は縦に繋がっているなど基本的だが大事なところを理解できた。
  また、汎用ロジック IC の規格を知る事ができた。Vcc から GND へ流れるようになっているので、ここを繋げないと使えない。どの汎用ロジック IC も、接続部が片方に 7 つあり両方で 14、そのうち 2 部が電流確保に使われるので、残りの 12 に 4 つのゲートが割り当てられているということが解った。

調査課題

下記の各項目について調査し、その結果を報告せよ。

(a) 2 変数の論理関数は全部で 16 種類ある。何故 16 種類になるか説明せよ。また、2 変数の論理関数を 16 種類すべて列挙し、否定 (NOT)、論理積 ((AND および論理和 (OR) のみを用いて表現せよ。
2 変数の論理関数ということはには $f(A, B)$ ということである。2 入力の組み合せは 4。

この入力にもそれぞれの組み合せがあるので $2^4 = 16$ なので、全部で 16 種ある。
16 種すべての論理関数を表 6 に示す。

A,B	00	01	10	11	式	
	0	0	0	0	$f = 0$	
	0	0	0	1	$f = AB$	AND
	0	0	1	0	$f = A\overline{B}$	禁止
	0	0	1	1	$f = A$	
	0	1	0	0	$f = \overline{A}B$	禁止
	0	1	0	1	$f = B$	
	0	1	1	0	$f = A \oplus B$	XOR
$f(A, B)$	0	1	1	1	$f = A \vee B$	OR
	1	0	0	0	$f = A \downarrow B$	NOR
	1	0	0	1	$f = A \Leftrightarrow B$	対等
	1	0	1	0	$f = \overline{B}$	
	1	0	1	1	$f = A \leftarrow B$	含意
	1	1	0	0	$f = \overline{A}$	
	1	1	0	1	$f = A \rightarrow B$	含意
	1	1	1	0	$f = A \mid B$	NAND
	1	1	1	1	$f = 1$	

表 6: 論理関数

(b) 今回の実験で学んだように、論理関数の表現は一意ではなく複数 (厳密には無数に) ある。実際に回路を実現する場合は、それらの関数の中で、できるだけ簡単な表現を採用すべきである。そのような簡単な論理関数表現を求める際、カルノー図と呼ばれる図表が用いられることがある。カルノー図とはどのような図表か調査し報告せよ。また、カルノー図の使用方法を具体例を用いて説明せよ。

カルノー図とは論理式を簡略化するために使われる表。入力と出力を分けて書き、出力で 2^n で隣り合う式を抜き出すことで簡単な式を作るのに使う。これを使用することで、回路を組む際のゲートを少なくすることができる。
まずは真理値表より、真の場合は'1'、偽の場合は'0' と真偽をカルノー図に書き込む。そして'1' に注目する。この時、2^n になる'1' の固まりを囲む。隣合う'1' や、四角の形になるものが多い。この囲まれた'1' の固まりから論理式を割り出すことができる。
また、カルノー図の特徴として'00'、'01'、'11'、'10' と並んでいるのが解る。これは、隣同士の真偽値が 1 だけ違うようにするためである。
例として真理値表を表 7、カルノー図を図 6 に示す。

A	B	C	f
0	0	0	0
0	0	1	0
0	1	0	0
0	1	1	1
1	0	0	0
1	0	1	1
1	1	0	1
1	1	1	1

表 7: 真理値表

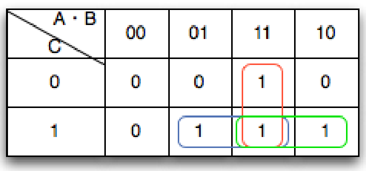

図 6: カルノー図

よって、この論理式は
$f = B \cdot C + A \cdot B + A \cdot C$
となる。

(c) 実用的なディジタル回路として半加算器および全加算器がある。これらの回路はどのような回路か調査し、それぞれの真理値表を示せ。また、各真理地表を用いて、それらの機能や特徴を説明せ

よ。さらに、それぞれの回路図の一例と、各回路図が実現している論理関数を示せ。

- 半加算器
  2 進数の最下位の演算で使われる加算機。AND、OR、NOT の組み合せで作られる。入力 A,B、出力 S、桁上げ出力 C がある。演算の結果で桁上がりがあると C へ出力され、この C は全加算機へ入力される。半加算器の真理値表は表 8、回路図は図 7 へ示す。
- 全加算器
  2 進数の最下位以外の演算で使われる加算機。2 個の半加算器と 1 個の OR からなる。入力 A,B、桁上がり入力 X、出力 S、桁上がり出力 C がある。半加算器の桁上がり出力は、ここの桁上がり入力 X へ入力される。全加算器の真理地表は表 9、回路図は図 8 へ示す。

A	B	C	S
0	0	0	0
0	1	0	1
1	0	0	1
1	1	1	0

表 8: 半加算器の真理値表

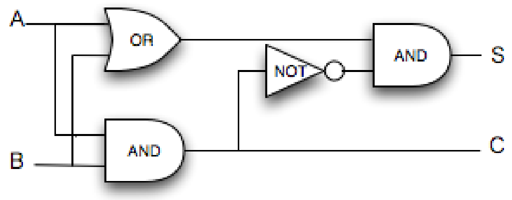

図 7: 半加算器の回路図

A	B	X	C	S
0	0	0	0	0
0	0	1	0	1
0	1	0	0	1
0	1	1	1	0
1	0	0	0	1
1	0	1	1	0
1	1	0	1	0
1	1	1	1	1

表 9: 全加算器の真理値表

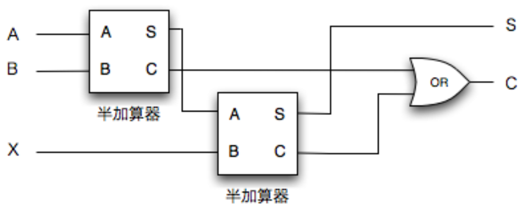

図 8: 全加算器の回路図

感想

実験は楽しくて早く終わったのにレポートは長くて大変でした。調査課題は理解しづらかったです。単純に文書作成するならこんなに時間はかかりませんでしえた。LaTex はエラーばっかりだしてきてくれます。溜息をついたのは一度や二度ではありません。他にも図を作成するのが大変でした。手書きだったらすぐに終わる作業だな…と思いながらやりました。

何とか期限内に出せてよかったです。次回はもう少し余裕を持ってやろうと思います。

参考文献・URL

- LaTex コマンドシート一覧 http://www002.upp.so-net.ne.jp/latex/
- LaTex-コマンド一覧 http://www1.kiy.jp/ yoka/LaTeX/latex.html
- http://www004.upp.so-net.ne.jp/tkonaka/bpse2/b2531.htm
- http://www.ei.fukui-nct.ac.jp/ sawai/LogicCircuit/Logic/Operations/Operations.htm
- http://ja.wikipedia.org/wiki/加算器